SOFTWARE ENGINEERING

DELIVERABLE-3

Group_4

Group Members:

1. Sadvik Kondadi
2. Sai Chandra Teja Akkala
3. Shashank Kodishala
4. Deepak Reddy Punuru
5. Deepu Gondhi
6. Padmaja Soma
7. Nithin Reddy Pinikesi
8. Sanjay Ramaswamy Adla
9. Thanwish Ram Pothugunta

1. Requirements

1.1 User Management

- **Account Creation & Management**: The system must support the creation and management of three user roles:
  - **Students**: Can self-register and access learning materials.
  - **Instructors**: Accounts are created by Admins, allowing them to manage courses and assignments.
  - **Admins**: Have full control over user and course management.
- **Role-Based Access Control**: Each user role should have predefined permissions ensuring restricted access to functionalities based on their role.

1.2 Course Management

- **Admin Functionalities**:
  - Create, edit, and delete courses.
  - Assign instructors to courses.
- **Student Functionalities**:
  - Enroll in available courses.
  - Access course materials such as lecture notes, videos, and assignments.

1.3 Assignment System

- **Instructors**:
  - Create and upload assignments for students.
  - Specify deadlines and instructions.

- **Students**:
    - Submit assignments through the platform.
    - Receive feedback and grades from instructors.

1.4 Assessment System

- **Quiz Functionality**:
    - Support for multiple-choice questions (MCQs) and True/False questions.
    - Auto-grading for MCQs to provide instant feedback.
    - Manual review for descriptive answers.

1.5 Content Management

- **Multimedia Support**:
    - Upload and organize learning materials, including PDFs, videos, and quizzes.
    - Provide structured content for easy student navigation.
- **Downloadable Materials**:
    - Students should be able to download course resources for offline access.

1.6 Grading System

- **Automated Grading**:
    - MCQs and True/False questions should be auto-graded to reduce instructor workload.
- **Manual Grading**:
    - Instructors can manually grade descriptive answers.
    - Provide feedback and grading comments for student improvement.

1.7 Customer Care

- **AI-Powered Chatbot**:
    - Integrate an LLM-based chatbot to assist students and instructors with common queries.
    - If queries cannot be resolved, escalate to Admins for intervention.
    - Maintain a history of conversations for future reference.

Requirements Moved to phase -2

1. **Progress Tracking & Reports**
    - Basic student progress tracking

    Reason: The project was delayed due to a connectivity issue with the MongoDB Compass and MongoDB Atlas. Moreover, some bugs were encountered in Content Management.

Development Phases:

5.1 Phase 1: Core Functionalities (MVP Development)

Objective: Establish a functional platform with essential features for user management, course management, and basic assessments.

1.User Management

• Student, Instructor, and Admin account creation & management

• Role-based access control

2. User Profile Update

• Profile editing options

• Private and Public mode for user information

3. Course Management

• The platform includes functions from Admin users to generate courses and administer course creation and modification and deletion processes.

• Students: Enroll in and access course materials

4. Assignment System

• Instructors create assignments

• Students submit assignments online

5. Assessment

• Basic quiz functionality (MCQs & True/False)

• Auto-grading for MCQs

6. Grading System

• Auto-grade MCQs

• The system uses manual grading processes for descriptive responses while providing feedback to students.

7. Customer Care

• LLM-based customer service for common queries

5.2 Phase 2: Advanced Learning Features

The main goal focuses on developing personalized recommendation systems with improved assessment tools supported by interactive enhancements.

8. Progress Tracking & Reports

• Basic student progress tracking

9. Recommended Courses for Specialization

• The system will create student recommendations that depend on the courses students have enrolled in.

10. Professor Recommendation System

• Educational data should receive evaluation scores based on the assessments from students in class.

11. Payment & Subscription Management

• Pay-training courses and the option to subscribe to programs are available through this platform.

12. Content Management

• The system enables users to upload along with organizing various multimedia platform content including videos PDFs and quizzes.

• Provide downloadable course materials

13. Assessment Enhancements

• Jumbled quiz questions for MCQs

• The system analyzes descriptive questions through similarity search methods for evaluation.

14. Attendance Tracking

• Track attendance for live sessions

• Generate attendance reports

15. Feedback & Survey System

• Collect student feedback on courses and instructors

5.3 Phase 3: AI-Powered Enhancements & Automation

The solution aims to use AI-based automation systems for evaluation tasks including assessment grading and administrative procedures.

16. RAG Implementation for Descriptive Questions

• Extract information from course material PDFs

• Transform data into vectors

• Load into a vector database for similarity-based evaluation

17. Grading System Enhancement

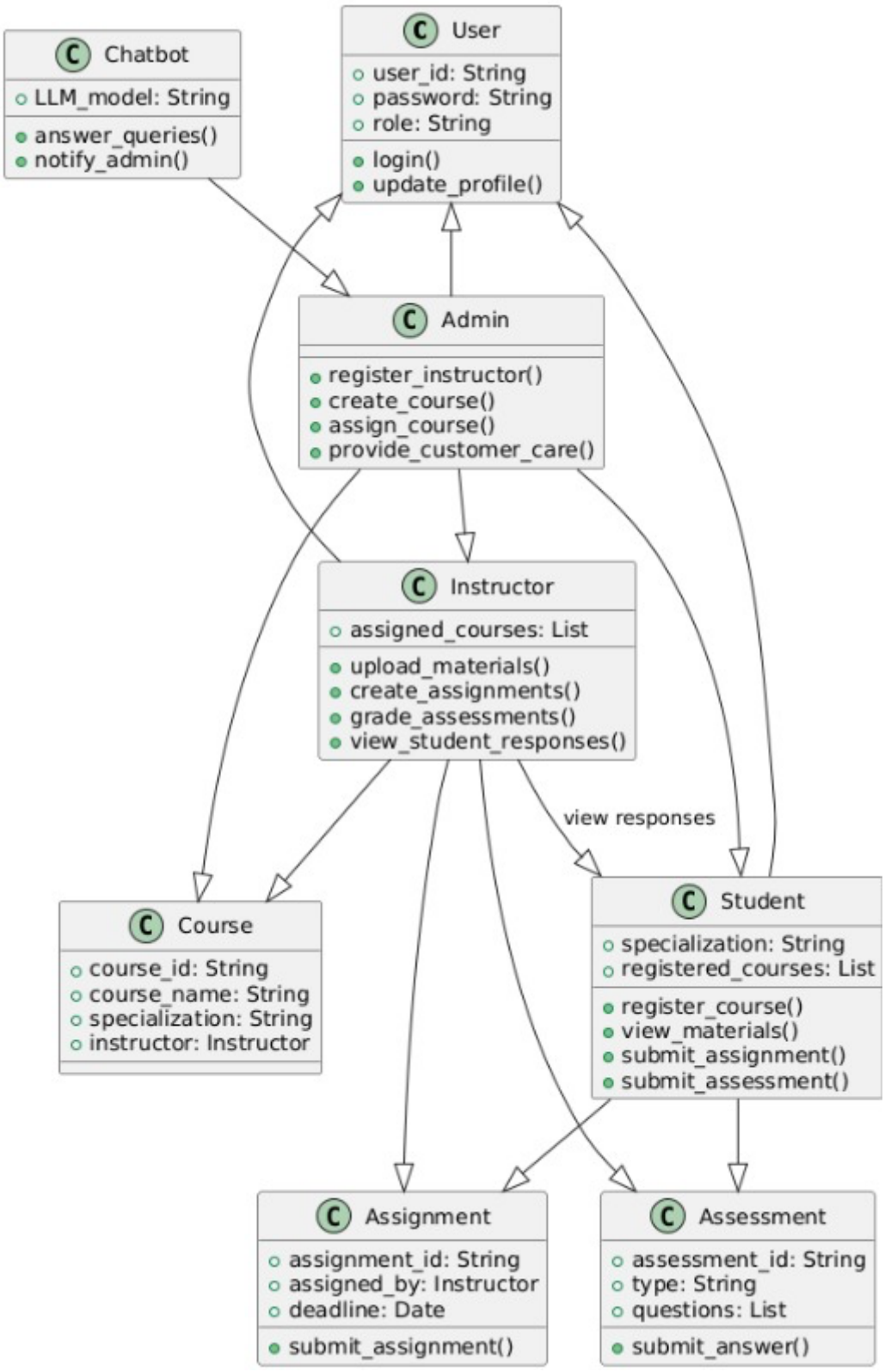
• AI-based auto-grading using cosine similarity for descriptive questions

18. Advanced Progress Tracking & Analytics

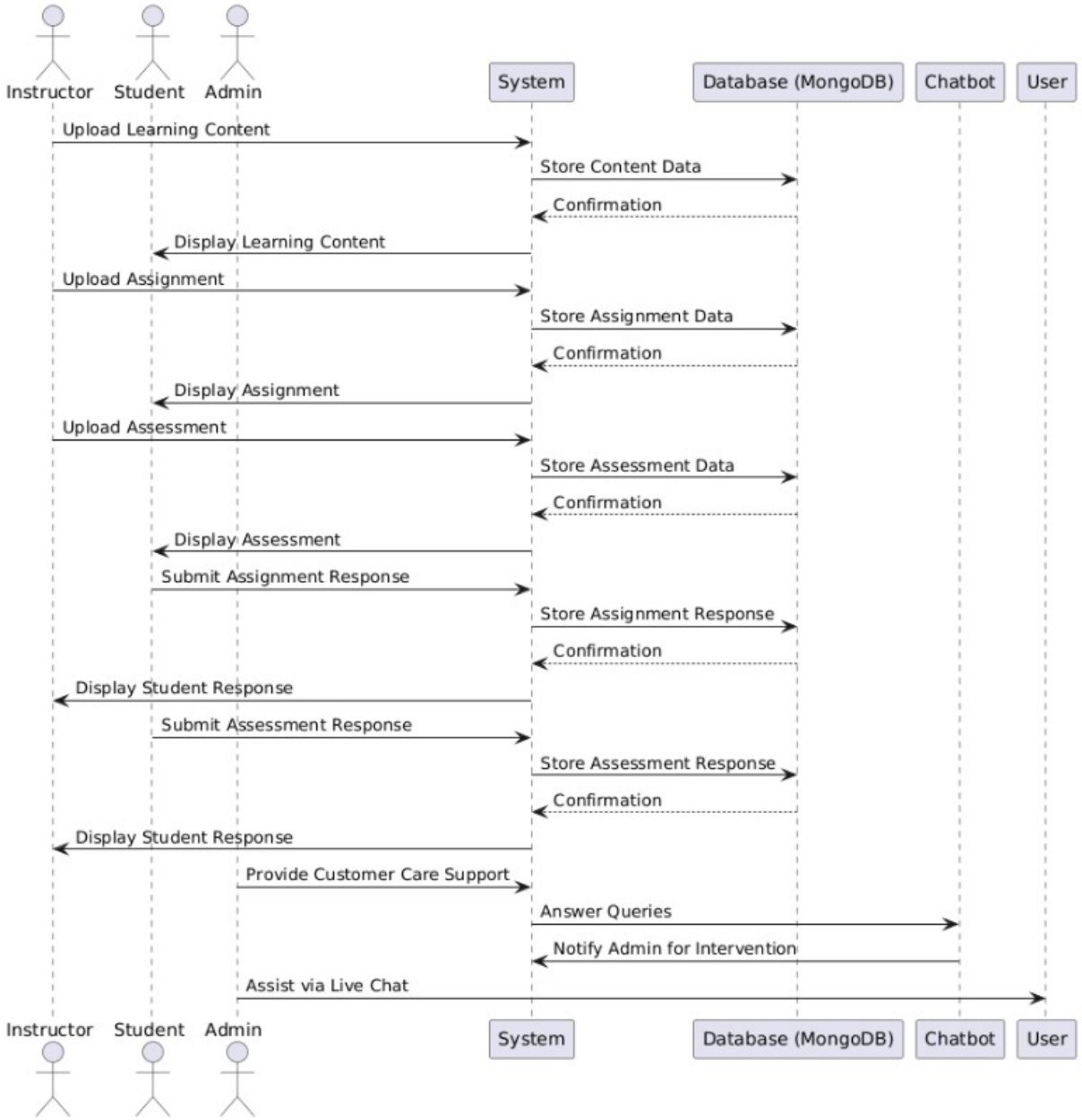
• The platform creates elaborate reports which benefit both student users and instructor

2. UML Design

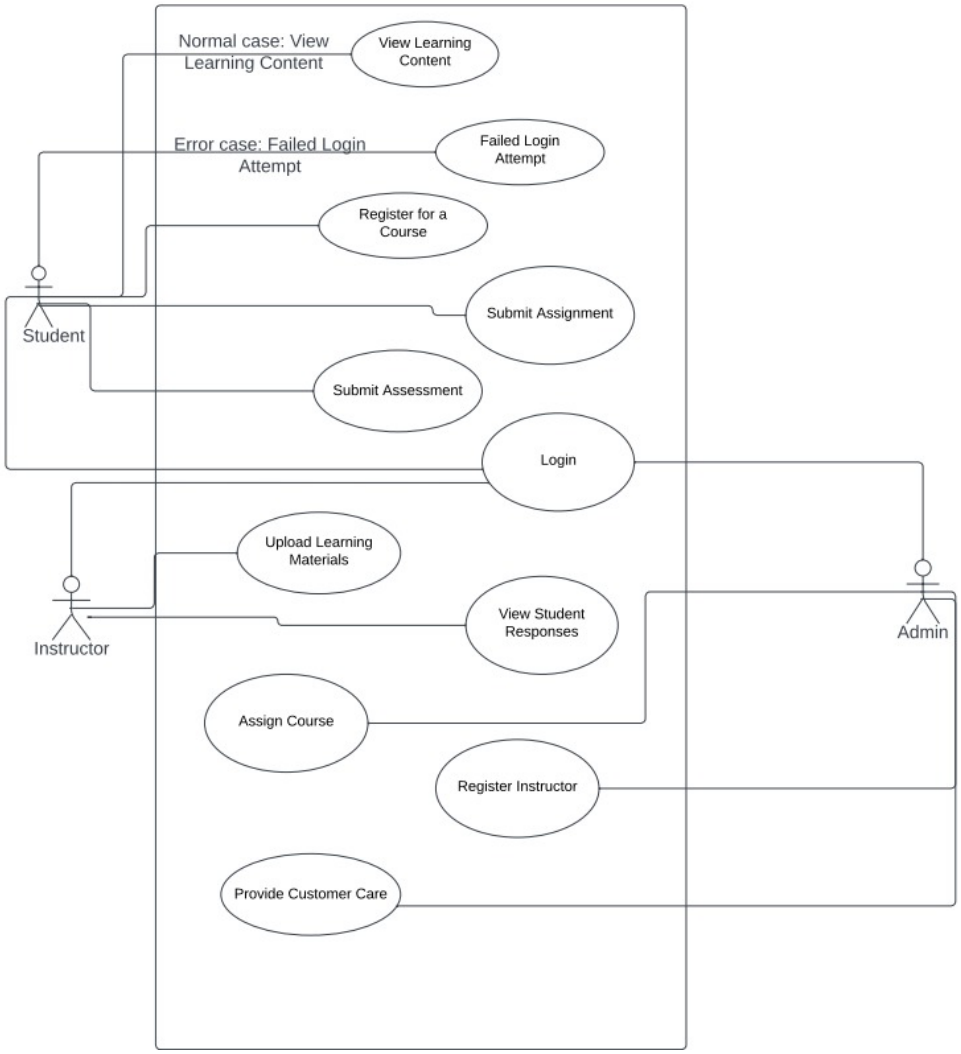
2.1 Class diagram



2.2 Sequence diagram



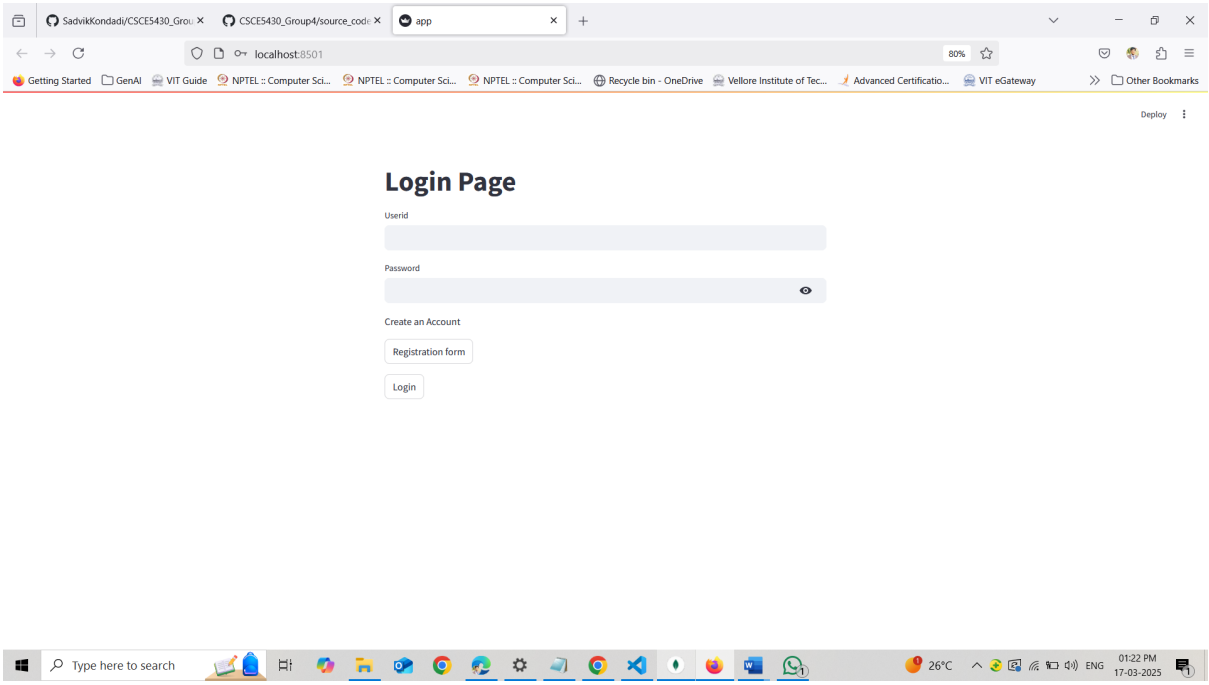
2.3 Use case diagram



3. Test Cases Covered

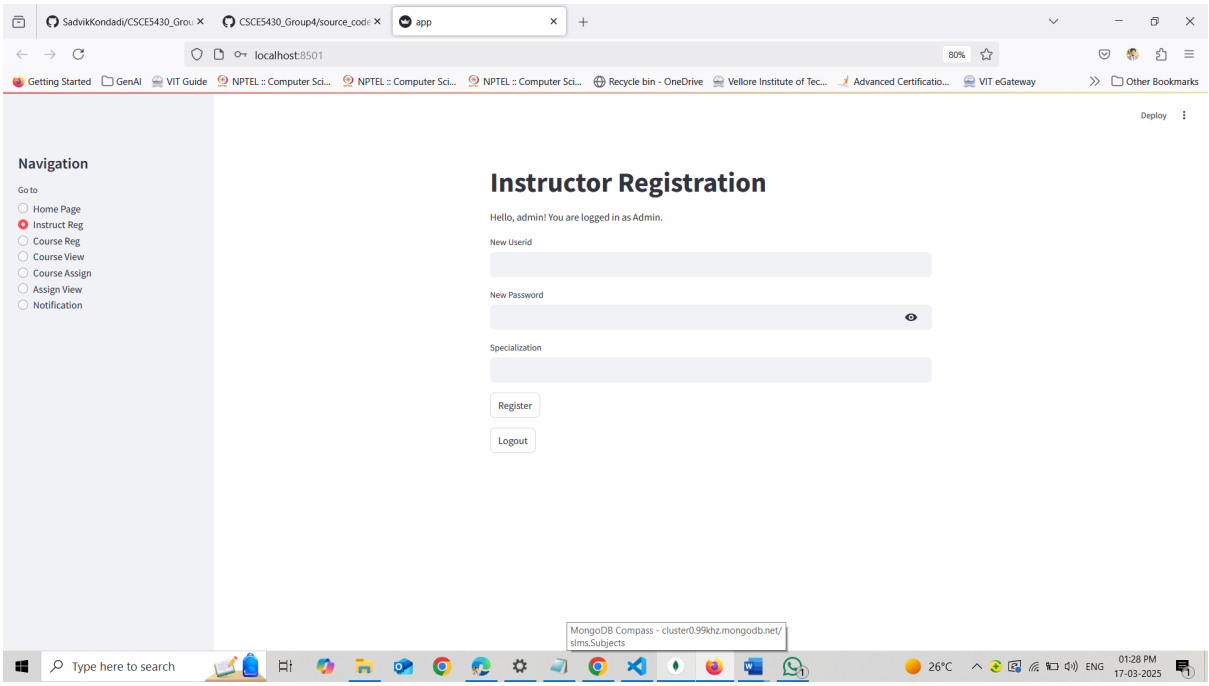
test_slms_ui_elements - Ensures login page elements exist.
test_slms_login_invalid_credentials - Tests login failure with incorrect credentials.
test_slms_login_success - Validates login with correct credentials.
test_slms_register_user - Tests the user registration process.
test_slms_dashboard_navigation - Checks navigation between sections.
test_slms_payment_process - Tests course payment feature.
test_slms_pdf_upload_download - Validates file upload & download.
test_slms_customer_care_chat - Tests chatbot functionality in customer support.
test_slms_logout - Ensures logout redirects users to login.

4. User Manual



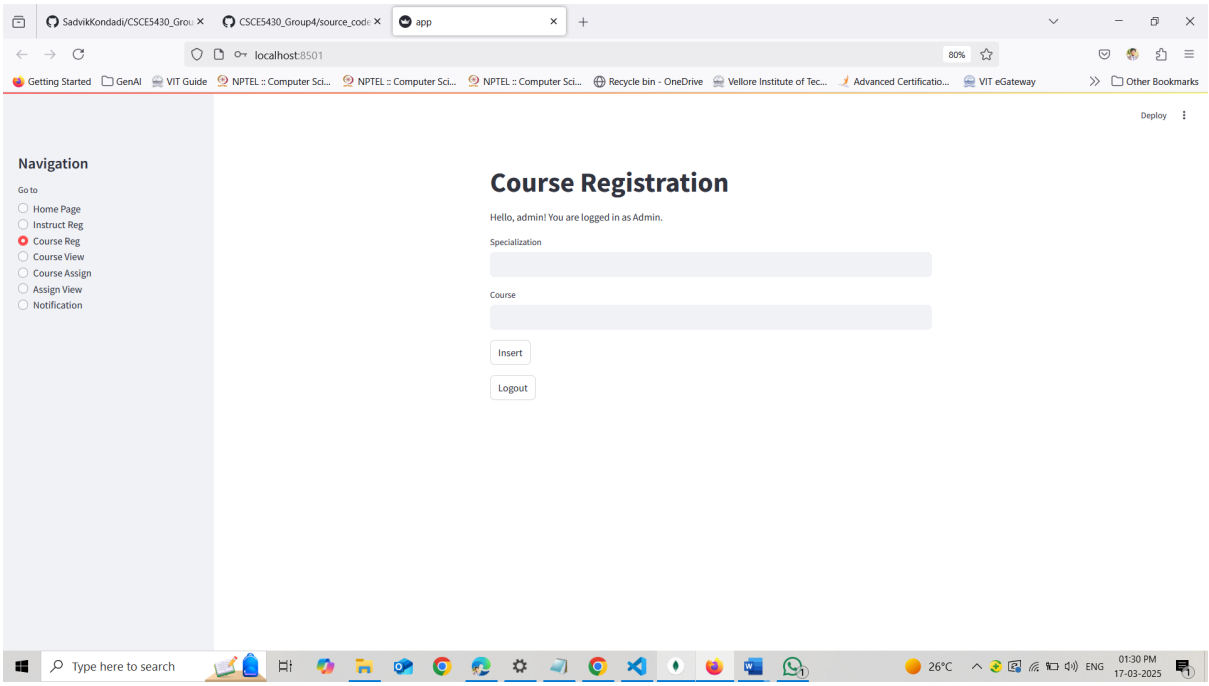
Login Page

The default browser will open to the login page, which includes fields for user ID and password input, a button that redirects to the registration form, and a button to log in. If the entered credentials match the document present in the MongoDB database, then the user will be logged in, and the MongoDB documents have a role field along with the credentials, which the user gives. The domain of the role field is ‘Admin’, ‘Student’, and ‘Instructor’.



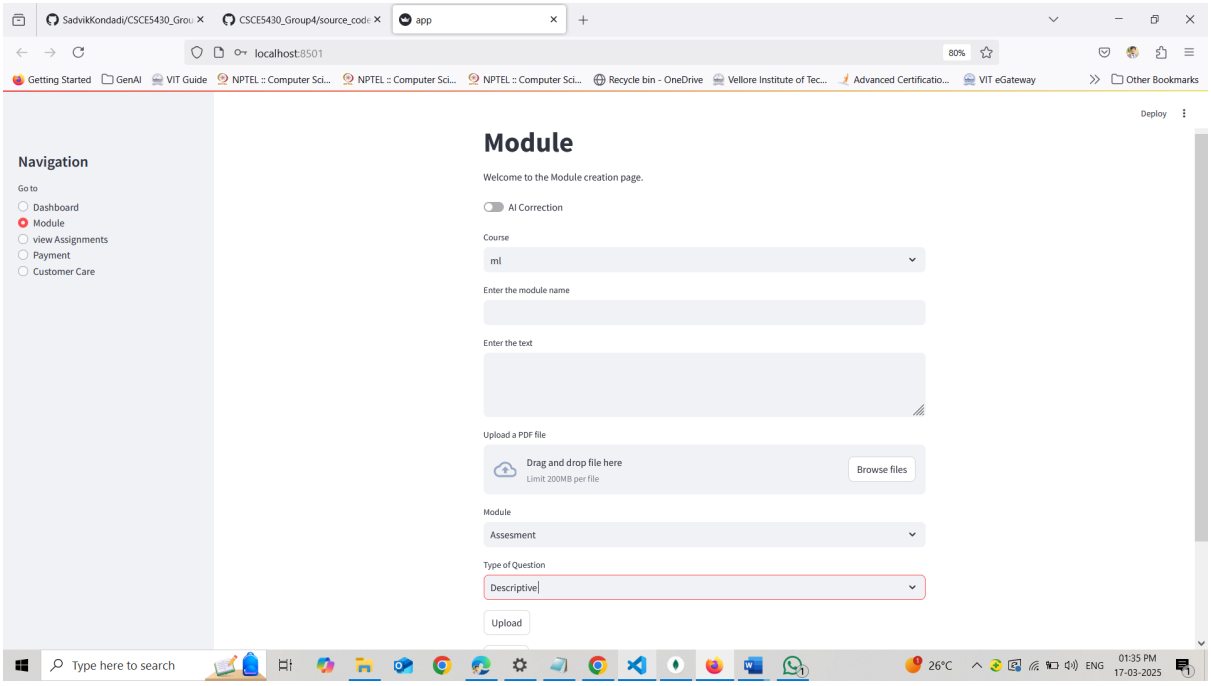
Admin View

Students create (Register) Student accounts, while Admin credentials are directly given by the code. The Admin registers Instructor Accounts. After logging in, both students and Instructors can update their profiles. The student can be registered by clicking the registration form button on the login page.



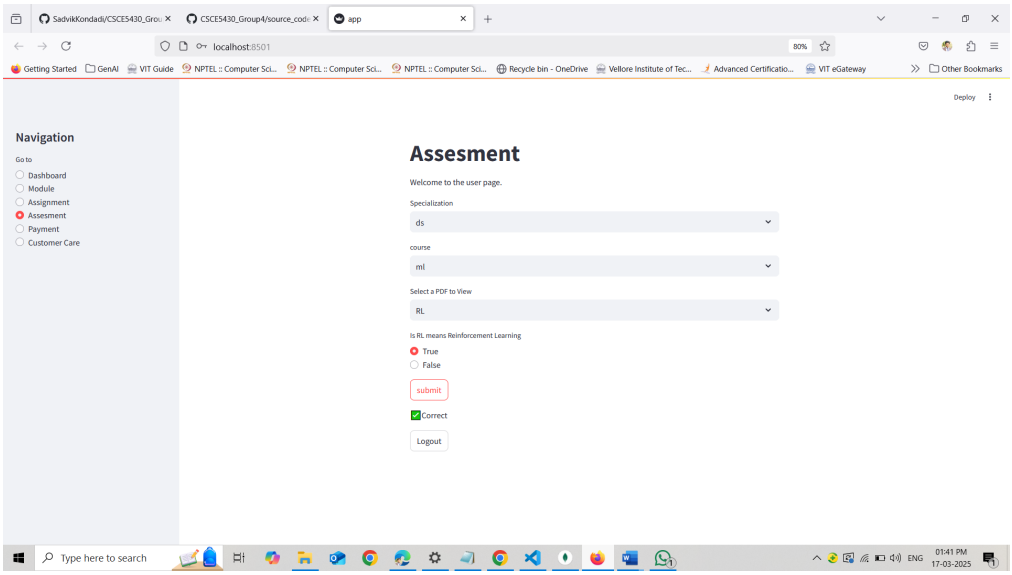
Course Management

The courses are registered by the admin under each specialization in the course registration tab. The courses are allotted to the Instructor by the Admin by filling the form in the Course Assign tab. The registered courses and the Instructor assigned to the course can be viewed in the ‘course view’ and ‘Assign view’ tabs respectively.



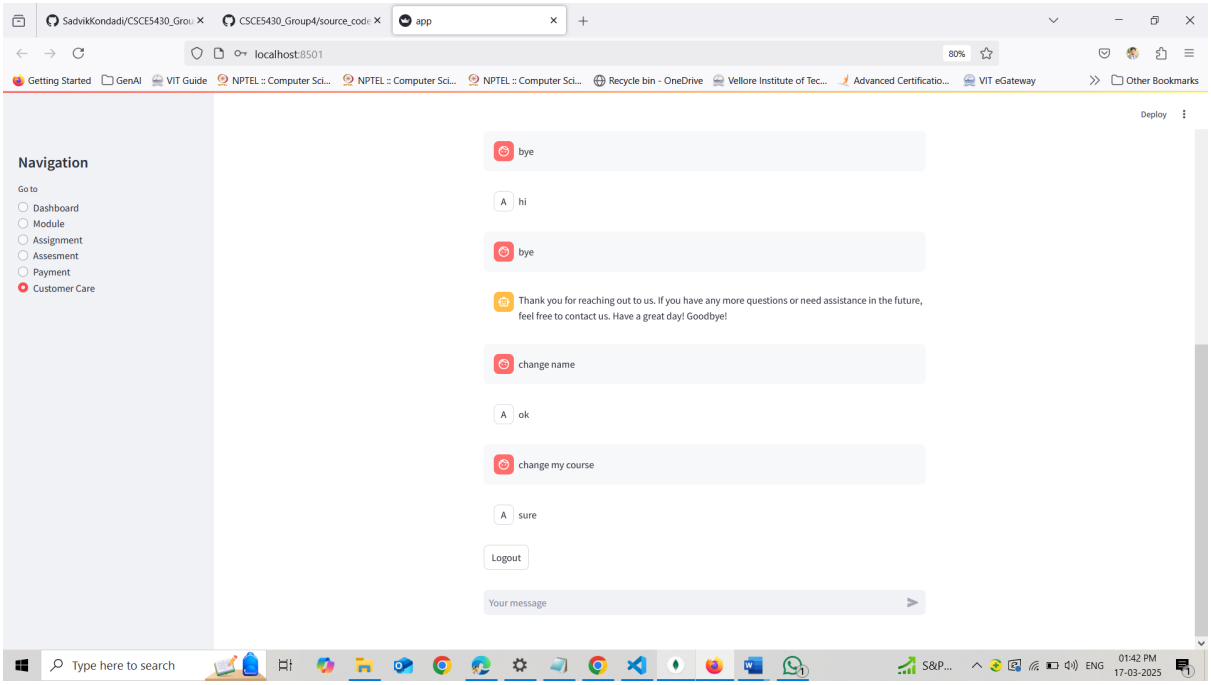
Assessment Instructor View

The materials, like learning content, assignments, assessments, are uploaded by the Instructor in the module tab and can be downloaded and displayed to the student in the Module (Learning Content), Assignment, and Assessment tabs respectively. The Assessment questions can be of four types. They are Descriptive Questions, MCQ Questions, More than one Answer MCQ Questions, and True or False Questions.



Assessment Student View

The Assignments are created by the lecturer, the answers will be uploaded to the student users in their respective modules.



Customer Care

Customer care is an LLM application-based chatbot that answers both students' and Instructors' questions, finds actionable items, and sends notifications to the Admin. Then the admin joins the chat, and the LLM is disabled, and the admin can perform the required tasks by communicating in the chat.

5.Instructions to Compile and Run the Code

- Download app.py code from Github, open in VS code and open terminal.
- The dependencies are installed by using the command ‘pip install -r requirements’ in the bash terminal.
- Run the program app.py by using the command 'streamlit run app.py' in the bash terminal.
- The default browser will open the Web UI.

6.Peer Review:

- We discussed regarding the chat between the admin and the student in customer care section.
- The other team told us to make the single interface where the reply for student comes with the same bot.
- So we accomplished it and made it to come as a chat between the admin and student when the question is actionable.
- They also asked whether the credit card details will be saved in database. So this section will be seen in next phase 2.

7. Reflection:

- The Implementation of Customer Care almost took 5 hours but in the end, it worked successfully.
- The Payment can be improved by using of bank third party to pay by credit Cards.
- The Document Upload has bugs but at last, the PDF view is Implemented.

8. Member Contribution:

Member name	Contribution description	Overall Contribution (%)	Note (if applicable)
Sadvik Kondadi	Admin account creation and management	11 %	
Sai Chandra Teja Akkala	Student and Instructor role based access control	11%	
Deepak Reddy Punuru	Course Registration	11%	
Deepu Gondhi	Course Assign view	11%	
Shashank Kodishala	Assignment system	11%	
Sanjay Ramaswamy Adla	Student Assignment submission	11%	
Padmaja Soma	User Profile update	11%	
Nithin Reddy Pinikesi	Assessment	11%	
Thanwish Ram Pothugunta	Customer care	11%	